PRINTER RUSH
(PTO ASSISTANCE)

Application : 10/752783 **Examiner :** Tung **GAU :** 2671

From: PAP **Location:** (IDC) FMF FDC **Date:** 10/18/05

Tracking #: EFM 10/752783 Week Date: 8/1/05

DOC CODE	DOC DATE	MISCELLANEOUS
☐ 1449		☐ Continuing Data
☐ IDS		☐ Foreign Priority
☐ CLM		☐ Document Legibility
☐ IIFW		☐ Fees
☐ SRFW		☐ Other
☐ DRW		
☐ OATH		
☐ 312		
☑ SPEC	1/6/04	

[RUSH] **MESSAGE:** Page 10, line 31 of the Specification has a missing Ser. No. Please advise.

Thank you

[XRUSH] **RESPONSE:**

See Misc Comm

DONE

INITIALS:

NOTE: This form will be included as part of the official USPTO record, with the Response document coded as XRUSH.

REV 10/04

[0050] Columns 670, 672, 674, 676, 678, and 679 illustrate the contents of registers R0 R1, R2, and R3, and buffers Z_H and Z_L, at each act of the command sequence 650. As can be seen, for the example where Z is less than Z_L, the final contents of registers Z_H is the previous value Z_L, while the final contents of buffer Z_L is the current value of Z.

[0051] The lines 662, 664, and 665 above highlight the usefulness of the conditional load (PLDC) and conditional store (PSTC) commands that are provided by embodiments of the present invention. These commands may be used to conditionally load and store values base on the content of a register or other location.

[0052] Figure 7 is an illustration of a graphics pipeline including embodiments of the present invention. Included are a host 705, geometry engine 710, rasterizer 715, shader front end 720, registers 725, texture filter and cache 730, shader back end 75, raster operations or ROP circuit 744, buffer interface 745, and graphics memory 750.

[0053] The host 705 receives primitives and textures from the AGP bus on lines or bus 702. The host provides the primitives to geometry engine 710, which processes them and outputs the result to the rasterizer 715. The rasterizer 715 provides fragments to the shader front end 720, which in turn couples to the registers 725. The shader front end 720 runs portions of the shader program and provides outputs to the texture filter 730 and shader back end 735. The texture unit 730 receives textures from the graphics memory 750 via the frame buffer interface 745, and provides them to the shader back end after optional filtering. The shader back end 735 also runs portions of the shader program and provides outputs to the raster operations circuit 740 and the shader front end 720. Specifically, for each pass through the shader, a number of fragments being operated on pass through the shader front end and shader back end once. When the passes that are required are completed, the shader provides an output to the raster operations circuit 740. Accordingly, there may be several shader passes occurring during each GPU pass.

[0054] The shader writes data to the frame buffer during a GPU pass via the raster operations circuit 740. Specifically, an arbiter circuit (not shown) in the rasterizer 715, shader 720, or raster operations circuit 740 selects data from the shader or raster operations circuit 740 and writes data to the frame buffer memory 750 via the frame buffer interface 745. This process includes what is referred to as memory position (or location) conflict detection mechanism (through interlocking), details of which can be found at copending United States patent application number 10/736,006, titled"."